

# 2회전 CORDIC을 이용한 QRD-RLS 알고리즘 구현

정희원 최민호\*, 송상섭\*\*

## QRD-RLS Algorithm Implementation Using Double Rotation CORDIC

Min-ho Choi\*, Sang-seob Song\*\* *Regular Members*

### 요약

본 논문에서는 변경된 Given회전을 이용한 QR 분해와 이를 기반으로 하는 RLS 알고리즘의 구현에 대하여 연구하였다. Givens 회전은 CORDIC(COordinate Rotation DIgital Computer) 연산을 반복 수행하여 구한다. QR 분해의 계산 시간을 줄이기 위해 1회의 Givens 회전에 필요한 CORDIC 연산의 반복 회수를 제한하였으며 보정계수를 계산할 때의 제곱근 계산을 없애기 위해 2회전 방법을 사용하였다.

**Key Words** : RLS; QR; CORDIC; Givens rotation.

### ABSTRACT

In this paper we studied an implementation of QR decomposition-based RLS algorithm using modified Givens rotation method. Givens rotation can be obtained with a sequence of the CORDIC operations. In order to reduce the computing time of QR decomposition we restricted the the number of iterations of the CORDIC operation per a Givens rotation and used double-rotation method to remove the square-root in the scaling factor.

### I. 서론

RLS(Recursive Least Squares) 알고리즘은 입력 데이터의 공분산 행렬의 고유치가 분산된 경우 수렴 특성이 좋아 자주 이용된다<sup>[2][3]</sup>. 계산속도와 안정성을 고려하여 QR 분해를 이용하여 공분산 행렬을 삼각화 하여 구현된 RLS 알고리즘의 성능을 보였다. Householder, Givens 회전 또는 Gram-Schmidt 변환 등을 통하여 행렬을 삼각화 할 수 있으며<sup>[1][16][17]</sup>, 표본 단위로 데이터를 처리하는 경우에는 Givens 회전이 적합하다. Givens 회전을 기반으로 하는 하드웨어 구현에 대해서는 Gentleman, Kung와 McWhirter 등에 의해 연구되었다<sup>[13][14]</sup>.

Volder에 의해 제안된 CORDIC(COordinate

Rotation DIgital Computer) 회전법<sup>[4][5]</sup> 행렬의 변환, 신호처리 및 초월 함수를 비롯한 다양한 수치 계산의 회로 구현에 널리 사용되고 있으며<sup>[18]</sup>, Givens 회전은 CORDIC을 이용하여 구현할 수 있으며 회전각의 계산에 소요되는 시간을 줄이는 것이 주요한 문제이다.

원래의 CORDIC 회전법을 이용하여 계산하는 경우, 요구되는 계산의 정밀도에 따라 미리 정해진  $b + 1$  회의 회전을 반복하여 원하는 각도로 벡터를 회전하지만, 본 논문에서는 한번의 Givens 회전을 수행할 때 반복되는 CORDIC 연산의 회수  $r$ 을  $1 \leq r \leq b + 1$  로 제한하는 방법을 사용하였다. 보다 정밀한 값이 필요한 경우는  $r$ 을 제한 범위

\* 전북대학교 대학원 전자공학과  
논문번호 : 030575, 접수일자 : 2003년 12월 26일

내에서 더 큰 값을 사용함으로써 해결할 수 있다.

논문의 나머지 부분은 II절의 QRD-RLS 알고리즘, III절의 Givens회전, IV절의 CORDIC을 이용한 Givens 회전 구현 그리고 마지막으로 V절의 결론으로 구성 되어 있다.

## II. QRD-RLS(QR Decomposition Recursive Least Squares) 알고리즘

### 1. RLS 알고리즘

이 절에서는 RLS 알고리즘을 QR 분해를 이용하여 표본단위로 재귀적으로 수행하는 과정을 기술한다.  $M$  차원 복소 벡터공간에 속하는  $\mathbf{x}[n]$  을 표본시간  $n$ 에  $M$ 개의 데이터 신호로부터 얻은 관측 벡터라 한다. 망각 계수  $\lambda$  ( $0 \leq \lambda \leq 1$ )에 의해 가중된 자승오차의 합을 최소화하는 조건 아래 원하는 신호  $y[n]$ 를 신호  $\mathbf{x}[n]$ 을 사용하여 추정할 수 있다.

$$\min_{\mathbf{w}[n]} \sum_{i=1}^n \lambda^{n-i} |y[i] - \mathbf{x}^T[i]\mathbf{w}[n]|^2$$

결국 이 문제는 잔류오차  $\epsilon[n]$ 의  $l_2$ -norm을 최소화하는 필터 계수  $\mathbf{w}[n]$ 를 구하는 것과 같다.

$$\|\epsilon[n]\| = \sqrt{\epsilon^H[n]\epsilon[n]},$$

$$\epsilon[n] = \mathbf{B}[n](\mathbf{y}[n] - \mathbf{X}[n]\mathbf{w}[n]) \in \mathbb{C}^n$$

$$\mathbf{X}[n] \equiv \begin{bmatrix} \mathbf{x}[1] \\ \vdots \\ \mathbf{x}[n] \end{bmatrix} \in \mathbb{C}^{n \times M}$$

$$\mathbf{B}[n] \equiv \text{diag}(\sqrt{\lambda^{n-1}}, \sqrt{\lambda^{n-2}}, \dots, 1) \in \mathbb{R}^{n \times n}$$

(1)

식 (1)에 정의된  $l_2$ -norm은 직교변환  $\mathbf{Q}[n]$ 을 적용해도 변하지 않으며, 가중된 입력 데이터 행렬  $\mathbf{B}[n]\mathbf{X}[n]$ 을 상삼각 행렬  $\mathbf{R}[n] \in \mathbb{C}^{M \times M}$ 로 변환하기 위하여 QR 분해를 적용한다.

$$\mathbf{Q}[n]\epsilon[n] = \begin{bmatrix} \mathbf{k}[n] \\ \mathbf{v}[n] \end{bmatrix} - \begin{bmatrix} \mathbf{R}[n] \\ \mathbf{0} \end{bmatrix} \mathbf{w}[n]$$

이 식으로부터 잔류 오차의 norm이 최소가 되는 조건은 다음의 관계가 성립하는 경우이다<sup>(1)</sup>.

$$\mathbf{R}[n]\mathbf{w}[n] = \mathbf{k}[n]$$

우선 QR 분해의 점화식을 유도하기 위하여 망각

계수 행렬  $\Lambda[n]$ 와 데이터 블록 행렬  $\mathbf{X}[n]$ 의 곱을 식 (2)로 표시한다.

$$\Lambda[n]\mathbf{X}[n] = \begin{bmatrix} \sqrt{\lambda}\Lambda[n-1]\mathbf{X}[n-1] \\ \mathbf{x}^H[n] \end{bmatrix}$$

$$\Lambda^2[n] \equiv \text{diag}\{\lambda^n, \lambda^{n-1}, \dots, 1\}$$

(2)

여기서  $\bar{\mathbf{Q}}[n-1]$ 를 다음 식으로 정의 한다.

$$\bar{\mathbf{Q}}[n-1] \equiv \begin{bmatrix} \mathbf{Q}[n-1] & \mathbf{0} \\ \mathbf{0}^H & 1 \end{bmatrix}$$

식 (2)에  $\bar{\mathbf{Q}}[n-1]$ 를 곱하여 다음 식을 얻는다.

$$\bar{\mathbf{Q}}[n-1]\Lambda[n]\mathbf{X}[n] = \begin{bmatrix} \sqrt{\lambda}\mathbf{R}[n-1] \\ \mathbf{0} \\ \mathbf{x}^H[n] \end{bmatrix}$$

(3)

Unitary 행렬  $\mathbf{Q} \equiv \hat{\mathbf{Q}}[n]\bar{\mathbf{Q}}[n-1]$ 은 식 (3)의 우변의 QR 분해를 수행하는 행렬  $\hat{\mathbf{Q}}[n]$ 을 구할 수 있으면,  $\Lambda[n]\mathbf{X}[n]$ 의 QR 분해에 직접 이용할 수 있다<sup>(3)(10)</sup>. 식 (3) 우변 행렬의 원이 모두 0인 행 벡터  $\mathbf{0}$ 는  $\hat{\mathbf{Q}}[n]$ 을 구하는 데 영향을 미치지 않기 때문에 결국  $\hat{\mathbf{Q}}[n]$ 를 구하는 것은 다음의 행렬을 QR 분해하는 것과 같다.

$$\begin{bmatrix} \sqrt{\lambda}\mathbf{R}[n-1] \\ \mathbf{x}^H[n] \end{bmatrix}$$

### 2. Givens 회전을 이용한 RLS 알고리즘의 QR 분해

앞의 2.1 절의 절차를 정리하면 식 (4)로 표시한다.

$$\mathbf{Q}[n] \begin{bmatrix} \sqrt{\lambda}\mathbf{R}[n-1] & \sqrt{\lambda}\mathbf{k}[n-1] \\ \mathbf{x}^H[n] & y^*(n) \end{bmatrix} = \begin{bmatrix} \mathbf{R}[n] & \mathbf{k}[n] \\ \mathbf{0}^T & e^*[n] \end{bmatrix}$$

(4)

$$\bar{\mathbf{R}}[n] = \begin{bmatrix} \sqrt{\lambda}\mathbf{R}[n-1] & \sqrt{\lambda}\mathbf{k}[n-1] \\ \mathbf{x}^H[n] & y^*(n) \end{bmatrix}$$

QR 분해를 한번 수행할 때, 즉,  $M$ 번의 복소 Givens 회전을 거쳐서 식 (4)의  $\mathbf{x}^H[n]$ 의 원들을 하나씩 0으로 만들어 가는 과정을 살펴본다. 우선 Givens 회전  $\mathbf{G}^{(1)}[n]$ 을  $\bar{\mathbf{R}}[n]$ 에 곱하면 그 결과는

$\mathbf{x}^H[n]$ 의 첫 번째 원 즉  $\overline{\mathbf{R}}[n]$ 의 마지막 행의 첫 번째 원을 0으로 만들며 다음 식이 된다.

$$\mathbf{G}^{(1)}[n]\overline{\mathbf{R}}[n] = \begin{bmatrix} c_1 & \mathbf{0}^H & s_1^* \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -s_1 & \mathbf{0}^H & c_1 \end{bmatrix} \times \begin{bmatrix} \sqrt{\lambda}r_{11}[n-1] & \sqrt{\lambda}r_{12}[n-1] & \cdots & \sqrt{\lambda}k_1[n-1] \\ \mathbf{0} & \cdots & \cdots & \vdots \\ x_1^*[n] & x_2^*[n] & \cdots & y^*[n] \end{bmatrix}$$

$$= \begin{bmatrix} r_{11}[n] & r_{12}[n] & \cdots & r_{1M}[n] & k_1[n] \\ \mathbf{0} & \cdots & \cdots & \vdots & \vdots \\ 0 & x_2^{(2)}[n] & \cdots & x_M^{(2)}[n] & y^{(2)}[n] \end{bmatrix}$$

회전의 파라미터  $c_1$ 과  $s_1 = \overline{\mathbf{R}}[n]$ 의  $(1, 1)$ 과  $(M, 1)$  원을 이용하여 구하며, 1행과 마지막 행인  $M+1$ 행의 1열을 제외한 나머지 열의  $M$ 개의 원들은 모두 변환된다. 같은 방법으로  $\sqrt{\lambda}r_{22}[n]$ 과  $x_2^{(2)}[n]$ 을 이용하여  $\mathbf{G}^{(2)}[n]$ 을 구하여, 마지막 행의 2번째 원을 0으로 만들고, 2행과 마지막 행( $M+1$ 행)의  $M-1$ 개의 원들을 역시 각각 변환한다. 결국  $\sqrt{\lambda}r_{ii}[n]$ 와  $x_i^{(i)}[n]$ 를 이용하여  $\mathbf{G}^{(i)}[n]$ 을 결정하고, 이것을 앞에서 구한 회전된

$$\begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \times \begin{bmatrix} 0 & \cdots & 0 & \sqrt{\lambda}r_{ii}[n-1] & \cdots & \sqrt{\lambda}r_{iM}[n-1] & \sqrt{\lambda}k_i[n-1] \\ 0 & \cdots & 0 & x_i^{(i)}[n] & \cdots & x_M^{(i)}[n] & y^{(i)}[n] \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \cdots & 0 & r_{ii}[n] & r_{i,i+1}[n] & \cdots & r_{iM}[n] & k_i[n] \\ 0 & \cdots & 0 & 0 & x_{i+1}^{(i+1)}[n] & \cdots & x_M^{(i+1)}[n] & y^{(i+1)}[n] \end{bmatrix}$$

$$c_i = \frac{\sqrt{\lambda}r_{ii}[n-1]}{r_{ii}[n]}, s_i = \frac{x_i^{(i)}[n]}{r_{ii}[n]}$$

$$r_{ii}[n] = [\lambda r_{ii}^2[n-1] + |x_i^{(i)}[n]|^2]^{\frac{1}{2}}$$

(5)

행렬  $\mathbf{G}^{(i-1)}[n] \cdots \mathbf{G}^{(1)}[n]\overline{\mathbf{R}}[n]$ 의  $i$ 번째 행과 마지막 행에 적용하여  $x_i^{(i)}[n]$ 를 0으로 변환한다 (7)(8)(9)

$\overline{\mathbf{R}}[n]$ 의 초기 조건을 실수의 단위 행렬로 하면, 즉  $\overline{\mathbf{R}}[0] = \delta \mathbf{I}$ ,  $\delta \ll 1$  이면, 대각원  $r_{ii}[n]$ 는 결국 양의 실수가 되어  $\overline{\mathbf{R}}[n]$ 의 대각원이 모두 양의 실수가 된다. 식 (5)의 연산을  $i = 1, 2, \dots, M$ 에 대해 반복함으로써  $\overline{\mathbf{R}}[n]$ 의 마지막 행의  $M$ 개의 원들을 모두 0으로 만들고, 그 결과 식 (4)의  $\mathbf{Q}[n]\overline{\mathbf{R}}[n]$ 은 상삼각 행렬이 된다.

이 과정에  $M(M+1)/2$ 번의 Givens 회전이

필요하다. 최종적인 직교화 행렬  $\mathbf{Q}[n]$ 은  $M$ 개의 Givens 회전 행렬의 곱이 되며 다음 식으로 표시된다.

$$\mathbf{Q}[n] = \mathbf{G}^{(M)}[n] \cdots \mathbf{G}^{(2)}[n]\mathbf{G}^{(1)}[n]$$

$$\mathbf{G}^{(i)}[n] = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & c_i[n] & \cdots & s_i^*[n] & & \\ & & & & 1 & & & \\ & & & \vdots & & \ddots & & \vdots \\ & & & & & & 1 & \\ -s_i[n] & & & & & & & c_i[n] \end{bmatrix}$$

(6)

이러한 과정을 거친 오차는  $\epsilon[n] = e[n]\alpha[n]$ 로 표시되며 여기서  $\alpha[n] = \prod_{i=1}^M c_i[n]$ 이다.

### III. Givens 회전

Givens 회전은  $n \times n$  대칭행렬  $\mathbf{A}$ 에 일련의 직교 변환  $\mathbf{G}$ 를 적용하여 이루어지며, 각 단계의 기본적인 연산은 순서쌍  $(p, q)$ ,  $1 \leq p < q \leq n$ 을 선택하여, 식 (7)의 좌변이 대각행렬이 되도록  $\cos, \sin$  쌍  $(c, s)$ 를 계산하고  $\mathbf{A}$ 를  $\mathbf{B} = \mathbf{G}^T \mathbf{A} \mathbf{G}$ 로 대체 하는 것이다<sup>(1)</sup>.

$$\begin{bmatrix} b_{pp} & b_{pq} \\ b_{qp} & b_{qq} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \quad (7)$$

식 (7)의 좌변의 대각화 과정으로부터 식 (8)을 얻는다.

$$b_{pq} = a_{pq}(c^2 - s^2) + (a_{pp} - a_{qq})cs = 0 \quad (8)$$

순서쌍  $(c, s)$ 은  $a_{pq} = 0$ 이면  $(1, 0)$  그렇지 않은 경우는 다음 식을 사용하여 정한다.

$$\tau = \frac{a_{pp} - a_{qq}}{2a_{pq}}, \quad t = s/c$$

식 (8)로부터  $t = \tan\theta$ 이며 2차식  $t^2 + 2\tau t - 1 = 0$ 의 근이다.

$$\begin{aligned} t &= -\tau \pm \sqrt{1 + \tau^2} \\ c &= 1/\sqrt{1 + t^2}, \quad s = tc \end{aligned} \quad (9)$$

2개의 근 가운데 작은 것을 선택하면  $|\theta| < \pi/4$ 를 보장하고 **A**와 **B**의 차이를 최소화 할 수 있음이 알려져 있다.

Givens 회전 **G**를 적용하는 목적은 비대각원 norm을 최소화하기 위함이다. 우선 변환전의 비대각원 norm을 다음의 식으로 표시한다.

$$\text{off}(\mathbf{A}) = \sqrt{\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}^2}$$

앞 식의 norm은 직교변환을 해도 변하지 않으며 이를 이용하여 다음 식을 얻는다.

$$\begin{aligned} a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2 &= b_{pp}^2 + b_{qq}^2 + 2b_{pq}^2 \\ &= b_{pp}^2 + b_{qq}^2 \end{aligned}$$

$$\begin{aligned} \text{off}(\mathbf{B})^2 &= \|\mathbf{B}\|_F^2 - \sum_{i=1}^n b_{ii}^2 \\ &= \|\mathbf{A}\|_F^2 - \sum_{i=1}^n a_{ii}^2 + (a_{pp}^2 + a_{qq}^2 - b_{pp}^2 - b_{qq}^2) \\ &= \text{off}(\mathbf{A})^2 - 2a_{pq}^2 \end{aligned}$$

결국 Givens 회전을 할 때마다 행렬 **A**는 대각행렬에 가까워진다. 위의 식으로부터  $b_{pq} = 0$  일 때 비대각원이 가장 크게 감소하는 것을 알 수 있다.

비대각원을 감소시키는 경우 Givens 회전을 계산하기 위하여 정확한  $(c, s)$  값이 필요하지는 않다. 근사값을 사용해도 직교성을 유지할 수 있으며, 이 때 비대각원은  $b_{pq} = da_{pq}$ ,  $0 \leq |d| < 1$ 으로 변환된다.

여기서  $d = c^2 - s^2 - 2\tau cs$ 에 식 (9)의 값을 대입하여 식 (10)를 얻으며  $|d|$ 는 계산의 정확도를 결정하는 파라미터이며 최대치가 작을수록 좋다.

$$d(\tau, t) = \frac{1 - 2\tau t - t^2}{1 + t^2} \quad (10)$$

#### IV. CORDIC을 이용한 Givens 회전 구현

##### 1. CORDIC 연산

CORDIC 연산의 기본 개념은 미리 정해진 기본각(基本角)의 집합  $\{\alpha_j\}$ 중의 각들을 이용한 일련의 회전을 수행하여 목표하는 회전각  $\theta$  만큼 vector를 회전하는 것이다<sup>(4)(5)</sup>.

$$\theta = \sum_{j=0}^b \sigma_j \alpha_j, \quad \sigma_j \in \{-1, +1\}$$

$$\begin{aligned} \alpha_j &\equiv \arctan(2^{-j}) \\ &\in \{45.0, 26.6, 18.4, 14.0, \dots\}, \\ j &\in \{0, 1, 2, \dots, b\} \end{aligned}$$

배율이 보정되지 않은 회전  $\mathbf{G}_u(\alpha_j)$ 를 식 (11)로 표시한다.

$$\begin{aligned} \begin{bmatrix} x_{j+1} \\ y_{j+1} \end{bmatrix} &= \begin{bmatrix} 1 & \tan\alpha_j \\ -\tan\alpha_j & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix} \\ &= \begin{bmatrix} 1 & \sigma_j 2^{-j} \\ -\sigma_j 2^{-j} & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix} \\ &\equiv \mathbf{G}_u(\alpha_j) \begin{bmatrix} x_j \\ y_j \end{bmatrix} \end{aligned} \quad (11)$$

회전되어 크기가 달라진 벡터의 크기를 원래의

크기로 조정하기 위해, 회전된 벡터에 보정계수  $K$ 를 곱하여  $b$  비트의 최종 결과를 얻는다.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = K \prod_{j=0}^{b-1} G_{\alpha_j}(\theta) = G(\theta) \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$K = \prod_{j=0}^{b-1} 1/\sqrt{1+2^{-2j}} \quad (12)$$

CORDIC의 2가지의 동작 모드 가운데 우선 벡터모드를 사용해서 입력 벡터  $(x_i, y_i)^T$ 를  $x$ 축을 향해 회전하여 벡터의 크기와 각도를 구한 후 회전모드에서 벡터모드를 적용해 얻은 각  $\theta_i$  만큼 입력 벡터  $(x_i, y_i)^T$ 를 회전한다. 식 (6)의 각각의 Givens 회전은 CORDIC의 회전모드를 적용하여 수행될 수 있으며, 이 때 사용되는 회전각은 CORDIC의 벡터모드를 적용하여 구한다.

2. 근사각을 사용하는 Givens 회전 계산

QRD-RLS 알고리즘에서 Givens 회전을 사용하는 이유는 각각의 새로운 입력 데이터 벡터  $x^H[n]$ 의  $M$ 개의 원을 0으로 만들기 위해서이다. 이 때 정확한 회전각  $\theta_i$ 을 사용하는 대신 다음 식을 만족하는 근사값  $\tilde{\theta}_i$ 을 사용하여 회전을 정의한다<sup>(6)(12)</sup>.

$$|x_i^{(i)}| = |d| \cdot |x_i^{(i)}|, \quad 0 \leq |d| \leq 1$$

감소계수  $d$ 는  $\tau = \sqrt{\lambda} r_{ii} / x_i^{(i)}$  과  $t = \tan \tilde{\theta}_i$ 를 이용하여 식 (10)로 앞에서 정의하였다. 회전각이 정확히  $\tan \theta_i = x_i^{(i)} / \sqrt{\lambda} r_{ii}$ 인 경우 당연히  $d = 0$ 이다. 구현을 쉽게 하기 위하여 사용 가능한 회전각을 기본각으로 제한하여  $\tilde{\theta} \in \{\alpha_j\}$ 로 한다. 식 (10)에  $t = \text{sign}(\tau) 2^{-j}$ 를 대입하여 식 (13)을 얻는다.

$$d(|\tau|, t) = \frac{1 - 2|\tau|2^{-j} - 2^{-2j}}{1 + 2^{-2j}} \quad (13)$$

$d(|\tau|, t) \leq 1/3$ 로 하는 경우,  $|\tau_i|$ 는 다음 식이 되고  $|\tau| \in [\tau_i, \tau_{i+1}]$ 이면  $j$ 값을  $j = i$ 로 선택한다. 즉, 가장 가까운 회전각의 근사

치를 기본각의 집합에서 선택한다.

$$|\tau_i| = \frac{1}{3} (2^i - 2^{-i+1})$$

식 (11)에,  $j$ 에 따라 변하는, 보정계수  $K_j$ 를 곱하여 배율이 보정된 회전  $G_K(j)$ 을 얻는다.

$$G_K(j) = K_j \begin{bmatrix} 1 & \tan \alpha_j \\ -\tan \alpha_j & 1 \end{bmatrix}$$

$$K_j = \frac{1}{\sqrt{1+2^{-2j}}}$$

보정계수  $K_j$ 를 쉽게 처리하기 위한 여러 가지 방법이 있으나<sup>(15)</sup>, 여기서는 제곱근의 계산을 없애기 위해, 앞 절에 정의한 기본각의 집합  $\{\alpha_j\}$ 에서  $j$ 를  $j+1$ 로 하여 원래의 회전각보다 작은 다음의 각을 사용하여 2번 회전 하였다. 예를 들면  $\alpha_j = 26.6^\circ$ 인 경우  $\alpha_j = 18.4^\circ$ 를 선택하여 2번 회전하며 결국 원래의 회전각인  $26.6^\circ$ 보다  $18.4^\circ \times 2 - 26.6^\circ = 10.2^\circ$  더 회전 한다. 결국 배율이 보정된 회전은 식 (14)로 표현된다.

$$G_{KD}(j) = K_j^2 G_K(j) G_K(j)$$

$$= \frac{1}{1 + 2^{-2j}} \begin{bmatrix} 1 - 2^{-2j} & \sigma 2^{-j+1} \\ -\sigma 2^{-j+1} & 1 - 2^{-2j} \end{bmatrix}$$

$$= \frac{1}{(1 + 2^{-2j})(1 - 2^{-2j})} \times$$

$$\begin{bmatrix} 1 & \sigma \frac{2^{-j+1}}{1 - 2^{-2j}} \\ -\sigma \frac{2^{-j+1}}{1 - 2^{-2j}} & 1 \end{bmatrix} \quad (14)$$

변경된 보정계수  $K_j^2 = 1/(1 + 2^{-2j})$ 는 이동-더하기 연산으로 다음 식을 이용하여 재귀적으로 구할 수 있다.

$$K_0^2 = \frac{1}{2},$$

$$K_j^2 = (1 - 2^{-2j}) \prod_{k=1}^j (1 + 2^{-2k+1})$$

$$\text{for } j > 0, \quad w = \log_2 \frac{b}{2j}$$

이동-더하기 연산수는  $j$ 가 증가함에 따라서 감소

하여  $j > b/2$  인 경우는 배율을 보정할 필요 없게 된다.

이상을 정리하면 원래의 경우에는 식 (12)을 이용하여  $b + 1$  회의 CORDIC 연산을 하여 정확한 회전각을 구하고 이 것을 사용하여 Givens 회전을 수행하였으나, 변경된 식 (14)을 이용하여  $b + 1$  이하의  $r$  회의 CORDIC 연산을 통하여, Givens 회전을 수행한다. 결국 CORDIC 연산의 회수를 줄여서 계산을 보다 빨리 마칠 수 있으며, 정확한 계산 값을 필요로 하는 경우는  $r$  의 값을 증가하여 최대  $b + 1$  이 되면 원래의 성능을 유지할 수 있다.  $j = j + 1$  로 함으로써 원래의 회전  $G_K(j)$  의  $t = \tan \alpha_j$  의 값은 바뀌게 되며 변경된 값을  $\tilde{t} = \sigma 2^{-j+1} / (1 - 2^{-2j})$  로 표시하고 이를 이용하여 변경된 감소 계수를 구한다.

변경된 감소계수는 식 (10)에  $\tilde{t}$  를 대입하여 구하며 식 (15)가 된다.

$$\begin{aligned}
 d(|\tau|, \tilde{t}) &= \frac{1 - 2|\tau|\tilde{t} - \tilde{t}^2}{1 + \tilde{t}^2} \\
 &= [(1 - 2^{-2j})^2 - 2^{2(-j+1)} - 2|\tau|(2^{-j+1})(1 - 2^{-2j})] \\
 &\quad \div [(1 - 2^{-2j})^2 + 2^{2(-j+1)}] \tag{15}
 \end{aligned}$$

식 (10)와 식 (15)를 비교한 결과를 그림 1에 보였다. 그림1의 점선이 원래 방법인 식 (10)을 사용하는 경우의  $d(|\tau|, t)$  이고, 실선이 2회전한 방법을 사용하는 경우인 식 (15)의  $d(|\tau|, \tilde{t})$  이다. 감소 계수는 상한이 0.587이다.  $|\tau|$  가 작은 경우를 제외 하면,  $|\tau|$  가 증가함에 따라 원래 방법의 감소계수인 식 (10)에 접근하는 것을 알 수 있다. 즉,  $|\tau|$  가 증가함에 따라 원래 방법과 2회전한 방법의 성능이 같게 됨을 알 수 있다.

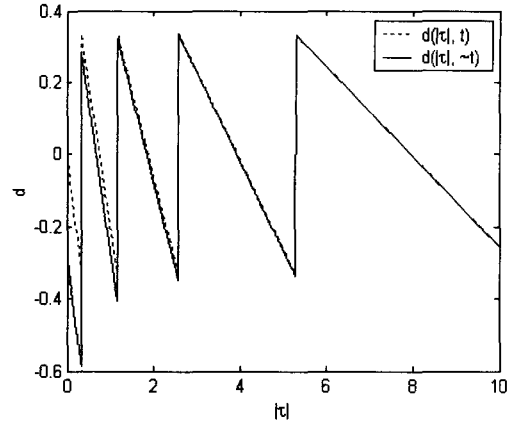


그림 5. 감소계수  $d$  와  $|\tau|$   
 Fig. 1 Reduction factor versus  $|\tau|$

### 3. 모의 시험 결과

$W = 2.9$ ,  $SNR = 30dB$ , 필터의 차수 10 을 사용하여 10회 반복한 모의 시험 결과를 그림2. 에 보였다. 수렴특성은 정확한 회전각을 이용한 Givens 회전을 사용하는 경우와 차이가 없으며, 예 측대로 1회의 Givens 회전을 수행할 때, 반복되는 CORDIC 연산의 회수  $r$  이 작아짐에 따라 초기의

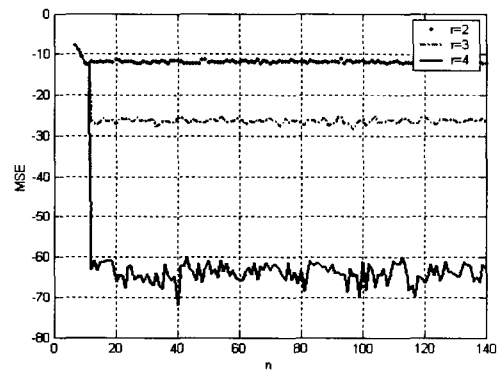


그림 2. 자승오차와 반복회수  
 Fig. 2 Mean squared-error versus iteration

수렴 특성에 상당한 영향을 미치며, 평균 자승 오차 의 분산이 증가하는 것을 관찰할 수 있다. 이는 필터의 차수를 고려하여 RLS 점화식의 수행 회수  $n$  에 따라  $r$  을 조정함으로써 알고리즘의 수렴 특성을 조절할 수 있다.

## V. 결 론

본 논문에서는 변경된 CORDIC을 이용하여 Givens 회전을 계산하고 이것을 이용하여 QR 분해를 수행하는 RLS 알고리즘을 구현하였다. 알고리즘을 하드웨어로 구현할 때는 복잡도와 연산속도를 고려하여 CORDIC에 사용하는 회전각도를 적절하게 선택하는 것이 중요하다. 즉, 정확한 회전각 대신 효율적으로 양자화 값을 구하는 것이 중요하며, 또한 가능한 한 작은 회수의 연산으로 계산의 속도를 증가하는 것 역시 고려해야할 사항이다.

이를 위하여 연산의 회수는 줄이고, 계급근 계산을 없애기 위해 사용하는 회전각은 기본각 가운데 원래의 연산에 필요한 다음의 각을 2번 사용하는 CORDIC 연산을 사용하여 RLS 알고리즘을 구현하고 변경된 파라미터의 영향을 분석하였다. 이 결과는 신호처리를 비롯한 다양한 분야에 적용될 것으로 기대한다.

## 참 고 문 헌

[1] G. H. Golub and C. F. Van Loan, Matrix Computations, Baltimore, MD: The John Hopkins University Press, 3rd Ed., 1996.

[2] S. Haykin, Adaptive Filter Theory, Prentice Hall, 3rd Ed., 1996.

[3] Manolakis, Ingle and Kogon, Statistical and Adaptive Signal Processing, McGraw-Hill, 2000.

[4] J. E. Volder, "The CORDIC Trigonometric Computing Technique," IRE Trans. Electronic Computers, vol. EC-8, no. 3, pp. 330-334, Sept. 1959.

[5] J. E. Volder, "The Birth of CORDIC," Journal of VLSI Signal Processing 25, 101-105, 2000.

[6] J. Götze, S. Paul, and M. Sauer, "An Efficient Jacobi-like Algorithm for Parallel Eigenvalue Computation," IEEE Trans. Computers, vol. C-42, no. 9, pp. 1058-1065, Sept. 1993.

[7] J. R. Cavallaro and A. C. Elster, "A CORDIC Processor Array for the SVD of a Complex Matrix," in SVD and Signal

Processing II -- Algorithms, Analysis and Applications, Elsevier Science Publishers Array B.V., Amsterdam, pp. 227-239, 1991.

- [8] R. McIlhenny and M. D. Ercegovac, "On the design of an online complex Givens rotation," Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on , Volume: 1 , 4-7, pp. 160 -164, Nov. 2001.
- [9] Thomas Kailath, Ali Sayed, Babak Hassibi, Linear Estimation, Prentice Hall, 2000.
- [10] K. R. Liu, S. -F. Hsieh, and K. Yao, "Systolic block Householder transformation for RLS algorithm with two-level pipelined implementation," IEEE Trans. Signal Processing, vol. 40 Issue 4, pp. 946-958, April 1992.
- [11] C. F. T. Tang and K. J. R. Liu, "A Fully Parallel Pipelined Systolic Array for MVDR Beamforming," TR\_91-89, Electrical Engineering Department and Systems Research Center University of Maryland College Park, MD.
- [12] Y. H. Hu, S. Naganathan, "An angle recording method for CORDIC algorithm implementation," IEEE Trans. Computers, vol. 42 Issue 1, pp. 99 -102, Jan. 1993.
- [13] W. M. Gentleman and H. T. Kung, "Matrix Triangularization by Systolic Arrays," in Real-Time Signal Processing IV, Proc. SPIE vol. 298, pp. 19-26, San Diego, CA, Aug. 1981.
- [14] J. G. McWhirter, "Recursive Least-Squares Minimization Using a Systolic Array," in Real Time Signal Processing VI, Proc. SPIE, vol. 431, pp. 105--112, San Diego, CA, Aug. 1983.
- [15] Jeong-A Lee and Tomas Lang, "A Constant-Factor Redundant CORDIC for Angle Calculation and Rotation," IEEE Transactions on Computers, Vol. 41, No. 8, pp 1016-1025, August, 1992.
- [16] J. W. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, 1965.
- [17] Lloyd N. Trefethen and David Bau,

Numerical Linear Algebra, Society for Industrial and Applied Mathematics, 1997.

- [18] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," IEEE Transactions on Computers, Vol. 41, No. 8, pp 1016-1025, August, 1992.

최민호(Min-ho Choi)

정회원

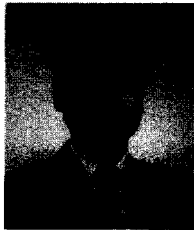


1991년 2월: 전북대학교  
전자공학과 (공학사)  
1993년 2월: 전북대학교 대학원  
전자공학과(공학석사)  
1993년 2월~1998년 6월:  
한국전자통신연구원  
1999년 2월~현재 전북대학교  
대학원 전자공학과 박사과정

<관심분야> 변복조, 신호처리, 부호이론

송상섭(Sang-Seob Song)

정회원



1978년 전북대학교 전기공학과  
학사  
1980년 KAIST 전기 및  
전자공학과 석사  
1990년 캐나다 마니토바  
대학교 전기 및 컴퓨터  
공학과 박사

1981년~현재: 전북대학교 전자정보공학부 교수

<관심분야> xDSL 모뎀, Home Network, 부호이론